

# Adaptació Web de “Memory Fields”

Xavier Jiménez Rodríguez-Arias

**Resum**—*Memory Fields* és una aplicació que ofereix una experiència multimèdia formada per col·leccions d'imatges i sons, mitjançant la qual, els usuaris poden crear diferents combinacions formades pels elements esmentats i emmagatzemar-les afegint una opinió o comentari. Degut a que l'aplicació requereix l'interacció d'usuaris, l'objectiu principal d'aquest projecte d'Enginyeria del Software ha estat el de crear una adaptació web en una pantalla de *Memory Fields* per tal d'oferir aquesta aplicació a un major nombre d'usuaris. Per dur a terme aquest projecte, s'ha realitzat una fase d'anàlisi de l'aplicació original i unes fases de disseny, implementació i proves de l'adaptació.

**Paraules clau**—Aplicació Multimèdia, Desenvolupament Web, Disseny Web, Enginyeria del Software.

**Abstract**—*Memory Fields* is an application that offers a multimedia experience by using image and sound collections. With this application, the users can create different combinations formed by the mentioned elements and save them alongside an opinion or a commentary. Due to the fact that the application needs user interaction, the main goal for this Software Engineering project is to create a single-page web adaptation based on *Memory Fields* to be able to offer this application to a larger number of users. In order to carry out this project, a phase of analysis of the original application and phases of design, implementation and tests of the adaptation had to be done.

**Index Terms**—Multimedia Application, Software Engineering, Web Design, Web Development.



## 1 INTRODUCCIÓ

EN l'actualitat existeixen més d'un bilió de pàgines web al món [1]. L'augment en els últims anys del desenvolupament web és degut a que el fet de crear una aplicació web proporciona avantatges com oferir una alta disponibilitat, accés desde diversos dispositius, compatibilitat entre sistemes operatius, entre altres. D'aquesta manera, es permet l'accés a un major nombre d'usuaris i, per tant, l'aplicació és més utilitzada.

D'aquesta idea neix l'objectiu principal d'aquest projecte d'Enginyeria del Software, crear una adaptació web en una sola pantalla de l'aplicació *Memory Fields* per tal de que un nombre major d'usuaris la puguin utilitzar.

### 1.1 Memory Fields

*Memory Fields* és una aplicació que, a través de diferents elements audiovisuals, ofereix una interacció amb la finalitat de permetre als usuaris la creació de memòries.

La interacció amb *Memory Fields* és simple i fàcil d'utilitzar. L'usuari pot navegar entre una col·lecció de pòsters de la Guerra Civil i una col·lecció de sons contemporanis de Catalunya i triar fins a dos elements de cada col·lecció. L'aplicació disposa de mescladors per incrementar o disminuir el volum dels sons i, alhora, alterar la visualització dels pòsters amb efectes. Finalment, l'usuari pot desar la combinació actual de pòsters i sons que, junt amb un comentari, formaran una memòria.

*Memory Fields* està pensat per funcionar en dues pantalles. Una pantalla tàctil des d'on l'usuari selecciona els elements i realitza les interaccions i una segona pantalla des d'on es visualitzen els pòsters i els efectes que es produeixen.

### 1.2 Elements de Memory Fields

Per realitzar les funcionalitats esmentades a l'apartat anterior, l'aplicació *Memory Fields* té els següents elements principals:

#### 1.2.1 Frames

Element utilitzat per mostrar i seleccionar els pòsters de la col·lecció.

#### 1.2.2 Decks

Element utilitzat per seleccionar els sons de la col·lecció.

#### 1.2.3 Sliders

Mescladors que regulen el volum dels sons i els efectes dels pòsters.

#### 1.2.4 Memory Flower

Element que representa la memòria d'un usuari. Conté els pòsters, sons i efectes disponibles en el moment que es crea, junt amb comentaris.

#### 1.2.5 Memory Field

Espai on es guarden totes les memòries dels usuaris.

#### 1.2.6 Carrusel

El carrusel és un element que apareix quan es vol seleccionar un pòster o so. Simplement mostra tots els pòsters o sons i permet seleccionar un.

- E-mail de contacte: [xavier.jimenezr@e-campus.uab.cat](mailto:xavier.jimenezr@e-campus.uab.cat)
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Fernando Luis Vilariño Freire (CVC), Dan Norton
- Curs 2015/16



Figura 1. Concepte de Memory Fields amb tots els elements [2].

### 1.3 Objectius

L'objectiu principal del projecte és el de crear una adaptació web en una sola pantalla de l'aplicació *Memory Fields*. Per arribar a aquest objectiu, s'ha realitzat una fase d'anàlisi de l'aplicació original i unes fases de disseny, implementació i proves de l'adaptació.

### 1.4 Estructura de l'article

Fins ara, en aquest article s'ha explicat una introducció, en què consisteix *Memory Fields* i l'objectiu principal d'aquest projecte. A continuació, a la secció 2 s'especificarà la metodologia de treball que s'ha seguit durant el projecte, així com la planificació inicial i les eines i llenguatges que s'han utilitzat. A la secció 3 s'explicarà la fase d'anàlisi que s'ha realitzat al llarg del projecte i es detallaran els requisits a complir. A la secció 4 s'esmentarà l'estructura de l'aplicació original i els dissenys realitzats per l'adaptació a partir d'aquesta. La secció 5 especifica tota la implementació que s'ha efectuat. Respecte a la secció 6, consisteix en una explicació de les proves que s'han dut a terme i finalment, la secció 7 resumeix les conclusions extretes un cop s'ha finalitzat el projecte.

## 2 METODOLOGIA

La metodologia, en l'àmbit del software, fa referència a la planificació i control de les diferents fases del projecte. Actualment existeixen diverses metodologies de treball, cadascuna d'elles sent utilitzada en projectes diferents.

En aquest projecte, ja que l'objectiu principal és crear una adaptació a partir d'una aplicació ja existent, s'ha seguit una metodologia de treball àgil adaptable als canvis, ja que a partir de l'anàlisi realitzat es produïen els dissenys i posteriorment la implementació.

El model seguit com a metodologia s'anomena espiral i és un model de cicle de vida del software que consisteix en iteracions formades per un conjunt d'activitats. No totes les iteracions comprenen les mateixes activitats sinó que es trien en funció del risc i de l'estat del projecte. Això permet un canvi de les activitats en cas d'aparició de problemes o errors durant el projecte i, per tant, permet una adaptació als canvis [3].

Per aquest projecte, les activitats que formaven els cicles de l'espiral normalment consistien en anàlisi, disseny

i implementació. L'activitat d'anàlisi estava formada principalment per entrevistes amb els tutors i/o creadors de l'aplicació original, investigacions del funcionament de l'aplicació i inspeccions del codi, l'activitat de disseny consistia a realitzar prototips visuals, diagrames i dissenyar noves possibilitats pels elements i, finalment, a l'activitat d'implementació es realitzava el desenvolupament del codi i les activitats relacionades amb la programació.

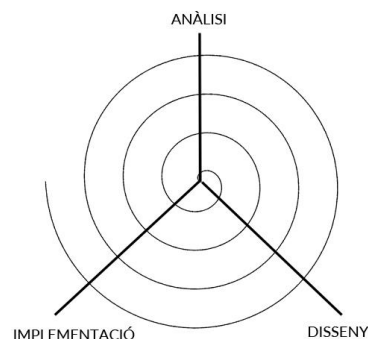


Figura 2. Representació del model en espiral d'aquest projecte.

### 2.1 Planificació

A l'inici del projecte es va realitzar una planificació tenint en compte les fases principals que formen un projecte d'Enginyeria del Software i el temps estimat per cada una. Amb aquestes dades es va construir un diagrama de *Gantt*, que és una eina gràfica on es pot veure la dedicació estimada per cada tasca [4]. En el diagrama es troben les etapes principals i la durada que es va estimar en setmanes:

Etapla	Febrer				Març				Abril				Maig				Juny			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Requisits																				
Disseny																				
Implementació																				
Proves																				

Figura 3. Diagrama de Gantt amb la estimació inicial del projecte.

### 2.2 Eines i llenguatges utilitzats

Al llarg d'aquest projecte s'han utilitzat diverses eines de software i llenguatges de programació.

Per realitzar la documentació, com els informes de seguiment, s'ha utilitzat *Microsoft Word 2016* [5] i per realitzar els diferents diagrames, *Cacao* [6]. A més, per tal de crear els dissenys i poder tractar imatges s'ha utilitzat *Adobe Photoshop CC* [7].

Per altra banda, a l'hora de desenvolupar, es va instal·lar un servidor local mitjançant un *framework* web de *Python* anomenat *Django* [8], ja que es tracta d'una web i, per tant, cal un servidor per poder treballar. Com a eina de creació i modificació de codi, s'ha utilitzat un editor de text anomenat *Sublime Text 3* [9]. S'ha treballat amb els llenguatges propis de desenvolupament web com *HTML*, *CSS*, *JavaScript* i *jQuery*.

Donada la necessitat mantenir tot el codi organitzat i

tenir la possibilitat de controlar les versions i els canvis realitzats, es va crear un repositori remot a *GitHub* [10] vinculat a un repositori local creat amb l'eina *Git for Windows* [11]. Aquesta eina, també va ser utilitzada a l'hora d'afegir i pujar els nous i/o modificats arxius.

Quan l'adaptació es trobava en un estat avançat, es va decidir pujar el codi a un servidor remot per tal que fos accessible per tothom i així poder rebre millor *feedback* dels tutors. Per dur a terme aquesta finalitat, es va utilitzar *OpenShift* [12], que es tracta d'una plataforma de *hosting* gratuït que permet la creació d'aplicacions web. D'aquesta manera, es va crear una aplicació amb el codi de l'adaptació i, a més, es va instal·lar una base de dades *MySQL* [13] per l'emmagatzematge de les col·leccions i les memòries i el mòdul *phpMyAdmin* [14] per gestionar-la.

### 3 FASE D'ANÀLISI

En un projecte d'Enginyeria del Software, la fase d'anàlisi té com a objectiu la recollida de tota la informació relacionada amb el projecte necessària per tal de poder definir correctament tots els requisits. En aquest projecte, s'ha adquirit tota la informació mitjançant diverses tasques.

La primera tasca que es va realitzar va ser la recopilació d'informació relacionada amb l'aplicació original. Dan Norton, un dels creadors del projecte original, va proporcionar informació referent a *Memory Fields*, com l'article publicat i conferències on es va presentar l'aplicació [2], [15], [16]. La lectura i comprensió d'aquestes fonts ha permès entendre la intenció i objectiu originals de l'aplicació per poder-ho traslladar a l'adaptació.

Com a segona tasca, trobem les entrevistes amb els creadors de *Memory Fields*. Aquestes han estat fundamentals per comprendre característiques de l'aplicació i les diferents decisions preses, així com rebre *feedback* sobre els progressos de l'adaptació.

Una altra tasca ha estat el fet d'interactuar amb la versió original de *Memory Fields*, poder provar les diferents funcionalitats ha permès entendre correctament com funciona l'aplicació i ha permès conèixer quines característiques principals ha de tenir l'adaptació.

Finalment, la tasca d'anàlisi i comprensió de codi ha estat una tasca iterativa que s'ha realitzat a mesura que el desenvolupament de l'adaptació avançava. Per adaptar les funcionalitats i/o afegir modificacions primer ha calgut comprendre com es troba estructurat i com funciona el codi a l'aplicació original.

Per tenir constància dels requisits extrets de les tasques anteriors, es va realitzar un document anomenat *Software Requirements Specifications* (SRS) que consisteix en un llistat dels requisits del projecte amb la seva identificació i descripció.

#### 3.1 Requisits funcionals

Els requisits funcionals són tots els requisits que defineixen una funció del software o dels components que el formen.

##### 3.1.1 Selecció de pòsters

Poder seleccionar un pòster des d'un *frame* i que es visualitzi.

##### 3.1.2 Selecció de sons

Poder seleccionar un so des d'un *deck* i que es reproduïxi.

##### 3.1.3 Eliminació d'un pòster

Poder treure un pòster prèviament triat.

##### 3.1.4 Eliminació d'un so

Poder treure un so prèviament triat.

##### 3.1.5 Producció dels efectes en els pòsters

Al triar un pòster i un so s'ha de produir un efecte en el pòster.

##### 3.1.6 Incrementar o disminuir el volum dels sons i/o els efectes dels pòsters

Mitjançant els *sliders* s'ha de poder graduar el volum del so i els efectes.

##### 3.1.7 Escriure un comentari

Poder escriure un comentari abans de crear una memòria.

##### 3.1.8 Guardar una memòria

Poder guardar una memòria amb els elements triats i un comentari.

##### 3.1.9 Afegir un comentari a una memòria existent

Poder afegir nous comentaris a memòries ja creades.

##### 3.1.10 Guardar les memòries en una base de dades

Poder guardar les memòries guardades pels usuaris en una base de dades.

##### 3.1.11 Aparició d'un salvapantalles en haver inactivitat

Després d'un temps d'inactivitat, que aparegui un salvapantalles amb el títol de l'aplicació.

### 3.2 Requisits no funcionals

Els requisits no funcionals són els requisits relacionats amb el software que no impliquen una funcionalitat directa.

#### 3.2.1 Disseny en una pantalla

Tots els elements de l'aplicació han de situar-se en una sola pantalla.

#### 3.2.2 Disseny responsive

El disseny ha de ser funcional en diferents resolucions de pantalla.

#### 3.2.3 Canviar el *Memory Field* i les memòries

Dissenyar un nou sistema de memòries i comentaris.

#### 3.2.4 Interacció dels elements amb *mouse*

Permetre la interacció de tots els elements amb el *mouse*.

#### 3.2.5 Adaptació operativa principalment amb el navegador Google Chrome

L'adaptació ha de ser funcional principalment en Google Chrome.

### 4 FASE DE DISSENY

La fase de disseny consisteix a realitzar dissenys i diagrames per determinar com funcionarà el *software* sense entrar en detalls d'implementació. En aquest projecte, pri-

mer s'ha realitzat una investigació i comprensió de l'estructura lògica i visual de l'aplicació original amb l'objectiu de determinar com realitzar l'adaptació.

#### 4.1 Estructura lògica de *Memory Fields*

En apartats anteriors s'ha descrit en què consisteix *Memory Fields*, els elements que formen l'aplicació i les funcionalitats que ofereix. A continuació, s'explicarà l'estructura lògica de l'aplicació original, informació que s'ha après a partir de les entrevistes amb els creadors i de les inspeccions de codi realitzades.

*Memory Fields* és una aplicació web desenvolupada amb els llenguatges propis de web com *HTML*, *CSS*, *JavaScript* i *jQuery*. És una aplicació local, és a dir, no realitza gairebé cap petició al servidor i tota la interacció i funcionalitat es produeix en el costat del client o usuari.

L'estructura lògica de l'aplicació ha estat creada mitjançant *BackBoneJS* [17], un framework ideat per la programació amb *JavaScript* que es basa en el patró de disseny d'aplicacions web *Model-View-Controller* (MVC). El patró MVC té com a objectiu dividir el codi de l'aplicació en tres parts interconnectades per tal de separar les dades, el tractament i la representació d'aquestes [18].

*BackBoneJS* és un framework, com ja s'ha comentat, dissenyat per ser utilitzat en aplicacions web d'una sola pàgina que tenen molta interacció en el costat del client. Per tant, és adient per l'aplicació *Memory Fields* i, també ho és per l'adaptació. D'aquesta manera, *BackBoneJS* desenvolupa el patró MVC diferenciant dues parts: el *Model*, que és l'encarregat de les dades i el que les proporciona quan es necessiten, i la *View*, que és la part visual de l'aplicació, la que reacciona a la interacció i la que escolta els canvis produïts per l'usuari.

##### Models and Views



Figura 4. Esquema de funcionament del MVC de *BackBoneJS*.

*BackBoneJS* també ofereix la possibilitat de definir *Collections*, que són agrupacions de *Models* iguals o similars. D'aquesta manera, es permet la modificació i es poden proporcionar tots els elements de l'agrupació alhora.

##### Collections

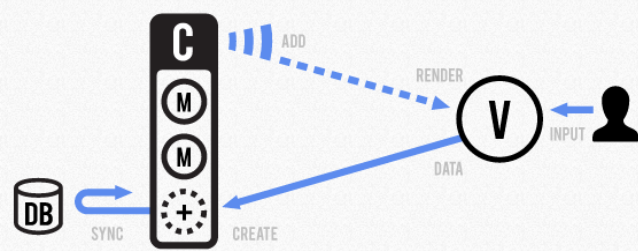


Figura 5. Esquema de funcionament de les *Collections* de *BackBoneJS*.

A *Memory Fields*, *BackBoneJS* s'utilitza de la següent manera:

Per cada un dels elements que formen l'aplicació, que han estat descrits en un dels apartats anteriors, es crea una estructura molt similar a una classe pel *Model* i una altra per la *View*. A *Model* es declaren les funcions i variables relacionades amb el comportament i funcionalitats de l'element, mentre que a *View* es declaren les funcions i variables relacionades amb la vista de l'element i la interacció que ofereix amb els usuaris.

Per poder crear un element, primer s'ha de declarar un *tag HTML*, com un contenidor, a la pàgina on l'element ha d'aparèixer. *BackBoneJS* utilitza *templates HTML* a l'hora de crear un objecte *View*, de forma que substituirà el *tag* contenidor amb el *template* definit a la classe. Per tant, tots els elements iguals utilitzaran el mateix *template*.

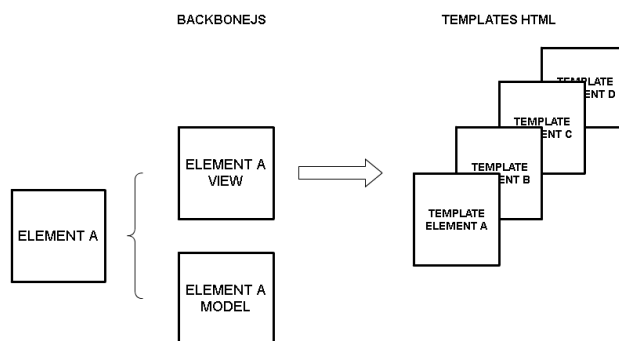


Figura 6. Esquema de funcionament de la declaració d'un element.

Cal afegir que a *Memory Fields* existeixen els elements anomenats *Mirror*. Els elements *Mirror* són elements pràcticament iguals que els que formen l'aplicació però que interessa diferenciar-los perquè realitzen alguna o varies funcionalitats diferents. L'exemple més clar es troba en els *frames*. Per una banda, els *frames* principals s'ocupen de mostrar els pòsters seleccionats, mentre que en els *frames Mirror* és on es produeix la interacció amb l'usuari, ja que és des d'on es poden seleccionar els pòsters.

#### 4.2 Estructura visual de *Memory Fields*

*Memory Fields* està ideat per funcionar en dues pantalles. Per tant, com a web, no és funcional, ja que es necessita la disposició dels dispositius adequats per poder utilitzar l'aplicació, i d'aquí surt el repte principal del projecte.

##### 4.2.1 Pantalla superior

La pantalla superior actua com a pantalla de visualització. En ella estiguin situats els dos *frames* que mostren els pòsters un cop seleccionats i els efectes que s'apliquen a ells.

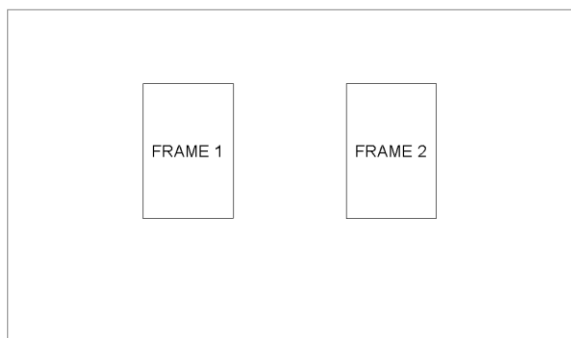


Figura 7. Representació de la pantalla superior de Memory Fields.

#### 4.2.2 Pantalla inferior

La pantalla inferior és la pantalla d'interacció i on es troben la majoria d'elements. Apareixen els dos *frames* utilitzats per seleccionar els pòsters, els dos *decks* utilitzats per seleccionar els sons i els tres *sliders* per regular el volum i els efectes. El *memory field* amb totes les *memory flowers* apareix un cop s'ha seleccionat almenys un pòster o so i es vol guardar una memòria.

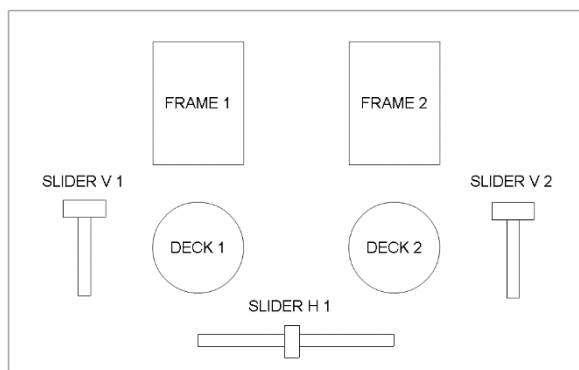


Figura 8. Representació de la pantalla inferior de Memory Fields.

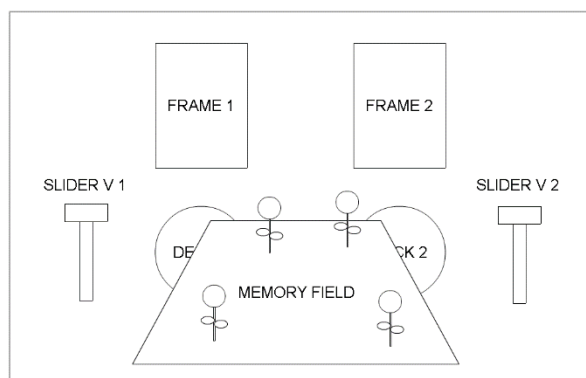


Figura 9. Representació de la pantalla inferior de Memory Fields amb el Memory Field obert.

#### 4.3 Disseny de l'adaptació de Memory Fields

Per realitzar el disseny de l'adaptació es van tenir en consideració diversos problemes que per tema d'espai calia solucionar.

Un dels problemes a simple vista és la reducció de l'espai que es produeix, concretament la meitat del espai original. Per tant, tots els elements havien de ser reduïts per tal que simplement apareguessin en la mateixa pantalla.

Això implicava un altre problema, la usabilitat. No només han de reduir-se sinó que han de ser igualment funcionals que a l'aplicació original i ser correctament usables per l'usuari amb el *mouse*. A més, els diferents elements no podien sobreposar-se els uns als altres.

Un altre problema que es va trobar involucrava el mòdul de creació de noves memòries. El fet que aparegui el *memory field*, la zona d'anotació de comentaris i els comentaris a la meitat de la pantalla no era una bona solució per l'adaptació. Es va produir el mateix problema amb el carusel de pòsters i de sons que apareix en seleccionar un *frame* o un *deck*. Per tant, va caldre pensar una ubicació per aquests elements que apareixen sota unes condicions.

Juntament amb Dan Norton, es va arribar a diversos dissenys que corregien els problemes mencionats anteriorment i altres que apareixien a mesura que la implementació avançava.

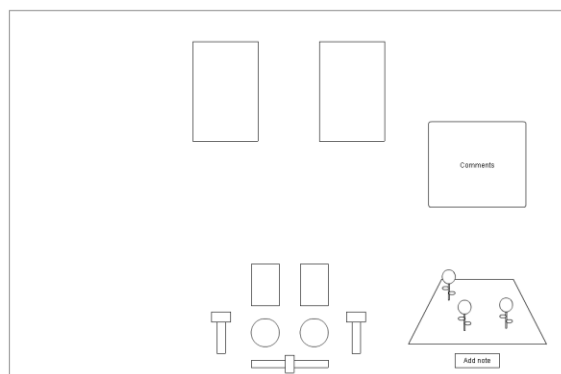


Figura 10. Disseny de l'adaptació de Memory Fields.

#### 4.4 Disseny de noves versions

A més del disseny a partir de l'aplicació original, va sorgir la idea d'utilitzar els elements que formen Memory Fields per crear noves versions diferents, amb temàtica relacionada o no. Entre diverses idees van destacar les següents:

La primera idea consistia en la creació d'un visor d'una col·lecció d'imatges de qualsevol temàtica centrat a interactuar amb una imatge. La idea es basava a poder seleccionar una imatge, poder fer-li zoom, permetre moure-la dins d'una àrea i guardar comentaris sobre ella.

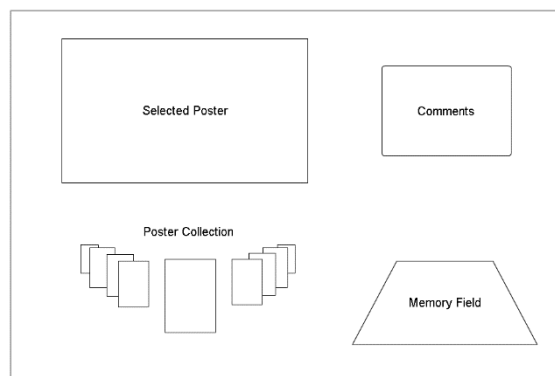


Figura 11. Disseny del visor d'imatges.



La segona idea es basava en la interacció amb sons. A l'aplicació apareixien diferents *decks* amb diferents sons en cada un d'ells als quals se li poden aplicar efectes de distorsió per tal de produir nous sons. A més, permetia la valoració d'usuaris i la possibilitat de guardar comentaris.

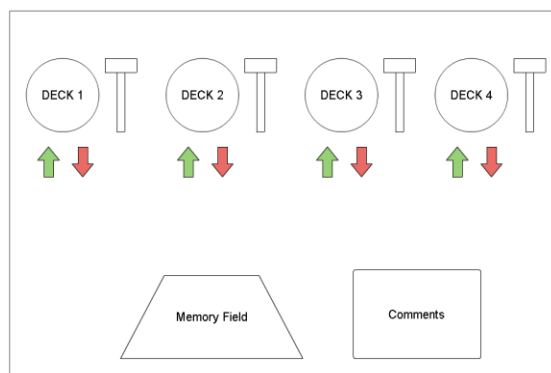


Figura 12. Disseny del mesclador de sons.

#### 4.5 Disseny de la base de dades

Una de les característiques de *Memory Fields* és la creació de memòries per compartir-les i formar un camp on diferents usuaris poden deixar la seva combinació d'elements junt amb una opinió. Per tant, un dels requisits és la creació d'una base de dades per poder guardar totes les memòries, a més dels altres elements audiovisuals que formen l'aplicació.

D'aquesta manera, es va realitzar un diagrama de les principals taules que calia tenir a la base de dades així com els camps principals.

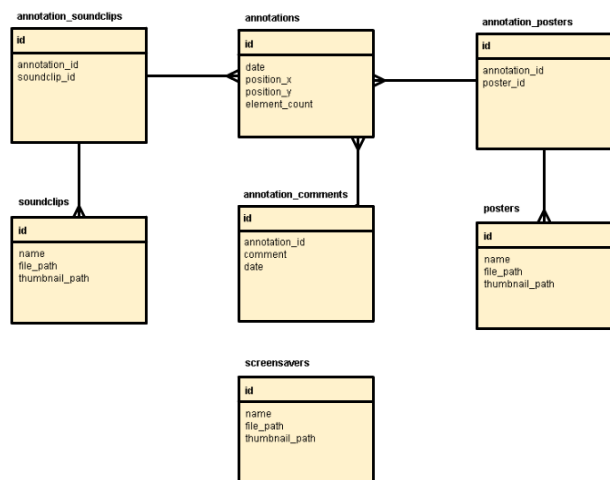


Figura 13. Diagrama Entitat-Relació de la base de dades.

Per una banda, es necessita una taula per cada un dels elements audiovisuals (pòsters, sons i salvapantalles). En aquestes taules cal un identificador, el nom de l'element, el directori on es troba l'arxiu per poder accedir a ell i el directori on es troba la miniatura per mostrar-la en el carrusel. Per altra banda, cal guardar les memòries i els elements que les formen. La taula *annotations* és utilitzada per em-

magatzemar un identificador de la memòria i la posició visual d'aquesta en el camp. Per guardar la informació contenida en la memòria s'utilitzen les següents taules: la taula *annotation\_comments* conté els comentaris escrits, la taula *annotation\_posters* els pòsters seleccionats i la taula *annotation\_soundclips* els sons triats.

## 5 FASE D'IMPLEMENTACIÓ

A la fase d'implementació d'un projecte d'Enginyeria del Software, l'objectiu principal és el de traduir els requisits adquirits durant la fase d'anàlisi i els dissenys realitzats a la fase de disseny a codi funcional. D'aquesta manera, la primera tasca que es va realitzar va ser la de comprovar l'efectivitat del primer disseny de l'adaptació proposat. Aquesta tasca va implicar diverses iteracions en les quals es va replantejar el disseny de l'adaptació i posteriorment es va tornar a implementar, ja que com s'ha especificat anteriorment s'ha seguit una metodologia de treball àgil i adaptable als canvis. Finalment, es va seguir el disseny esmentat a l'apartat 4.3.

### 5.1 Adaptació del elements

L'espai disponible per la col·locació dels elements es redueix, com a mínim, a la meitat, ja que l'objectiu és adaptar l'aplicació per tal que sigui operativa en una sola pantalla. A més, l'adaptació ha de ser *responsive*, és a dir, ha de poder adaptar-se a diferents resolucions de pantalla. Per tant, en resolucions baixes l'espai disponible es redueix encara més.

Per implementar el disseny, el primer pas ha estat la col·locació de tots els elements necessaris a la primera pantalla mitjançant l'arxiu *HTML* de la pàgina principal. Com ja s'ha comentat anteriorment, amb l'ús de *tags HTML* es defineixen els espais on se situaran els elements, que posteriorment seran creats amb els *templates* definits a l'estructura *View* de *BackBoneJS*. El fet de modificar la mida, l'espai o els *templates* dels elements ha implicat adaptar també algunes funcionalitats. Les modificacions realitzades en els diferents elements han estat les següents:

#### 5.1.1 Frames

Es necessiten quatre *frames* per realitzar la funcionalitat de mostrar i seleccionar pòsters. Els dos *frames* principals, els que mostren els pòsters, han estat col·locats, s'ha reduït la seva mida segons el disseny i s'ha realitzat una modificació del *template* per poder reduir també la mida dels pòsters que es mostren. Per altra banda, els dos *frames Mirror*, on l'usuari pot interactuar per triar els pòsters, també han estat posicionats i reduïts considerablement. A més, s'ha canviat el *template* per eliminar el nom del pòster que apareixia dins del *frame* en seleccionar-lo, ja que al reduir la mida no era llegible.

#### 5.1.2 Decks

A l'aplicació original es pot observar un cas similar a la col·locació dels *frames*. A la pantalla superior existeixen dos *decks*, tot i que ocults, i a la pantalla inferior dos *decks Mirror* funcionals. En aquest cas, només calen dos *decks* per

implementar la funció de poder seleccionar els sons i, per tant, s'ha modificat el *template* dels *decks* principals per tal de visualitzar-los com al disseny, s'ha adaptat el codi on es creen els elements per tal de fer-los funcionar sense necessitat d'un *Mirror* i s'han eliminat els *decks Mirror*. A més, s'han posicionat i reduït la seva mida seguint el disseny.

### 5.1.3 Sliders

Els *sliders* no utilitzen cap *template* HTML, ja que només són formats per dos elements. Per tant, simplement s'han posicionat i s'ha reduït la seva mida tal com apareix en el disseny. Tot i això, el fet de reduir la mida dels *sliders* va provocar un funcionament incorrecte a l'hora de regular el volum dels sons i els efectes dels pòsters. Per tant, s'ha modificat la funció per tal que els *sliders* verticals regulin correctament els sons i els efectes entre el 0% i el 100%, i el *slider* horitzontal canviï entre els sons seleccionats a l'esquerra i a la dreta.

### 5.1.4 Memory Field

En aquest apartat s'inclourà tot el mòdul de generació de comentaris, ja que es va canviar considerablement respecte a l'aplicació original. Inicialment, l'espai on es guarden les memòries o *Memory Field* apareixia a la meitat de la pantalla per sobre de tots els altres elements, com es pot veure a la Figura 9. Aquesta funcionalitat comportava diversos problemes d'usabilitat, ja que al superposar la resta d'elements s'evita la interacció de l'usuari amb ells un cop obert el *Memory Field*. A més, per accedir a la resta de memòries dels usuaris calia iniciar la creació d'una nova. Per tant, es va decidir canviar la funcionalitat d'aquest mòdul.

Primerament, s'ha posicionat i reduït la mida del *Memory Field* seguint el disseny realitzat. A l'adaptació, el *Memory Field* sempre es troba disponible, de forma que l'usuari pot accedir a altres memòries sense necessitat de realitzar cap altra acció prèvia. Per fer-ho possible, es va canviar el funcionament que permet crear una nova memòria. A l'adaptació, la selecció d'un element audiovisual qualsevol provoca l'aparició d'un botó per accedir a la creació d'una memòria. Llavors, al fer click, apareix la zona d'anotació per escriure el comentari i el botó per guardar la memòria.

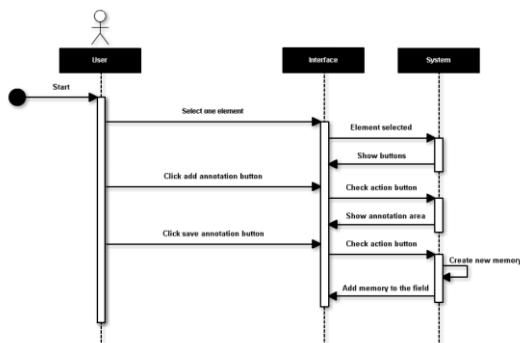


Figura 14. Diagrama de seqüència per la creació d'una memòria.

Per altra banda, un cop implementat el funcionament anterior, es va buscar solució a un altre problema. En el moment que es crea i es guarda una memòria, aquesta se situa en el *Memory Field* per tal que el propi o altres usuaris

puguin accedir a ella i veure els comentaris. El problema apareix quan el *Memory Field* s'omple de memòries i no hi ha espai per més. Per tant, l'objectiu era buscar una solució a aquest problema, ja que la idea principal de l'aplicació és permetre la interacció d'un gran nombre d'usuaris amb un nombre il·limitat de memòries.

Després de realitzar diversos dissenys i dues implementacions, es va decidir com a versió final deixar el *Memory Field* fix i permetre un desplaçament horitzontal a mesura que anés incrementant el nombre de memòries.

Per implementar-ho s'ha modificat el *template* del *Memory Field* afegint la zona on es col·loquen les memòries. Aquesta zona anirà augmentant la seva mida quan una nova memòria no trobi un lloc lliure per situar-se. La funcionalitat esmentada s'ha realitzat modificant la funció que guarda les memòries, la funció de col·locació de memòries sobre el *Memory Field* i la funció que detecta col·lisions entre memòries.

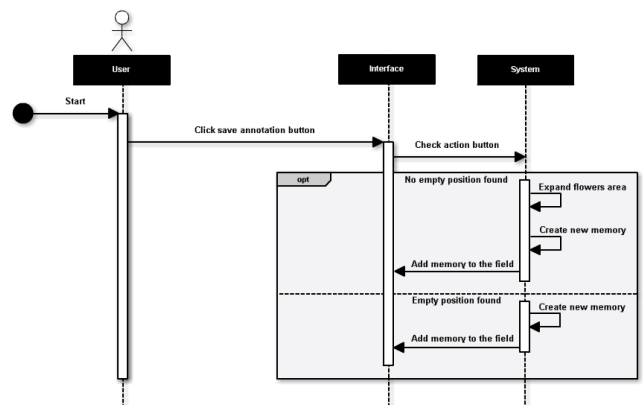


Figura 15. Diagrama de seqüència per la col·locació d'una memòria.

Per finalitzar les modificacions del mòdul de memòries, s'ha reduït la mida de les flors o *Memory Flowers* i s'han posicionat i també adaptat les mides de la zona d'anotació i dels comentaris que apareixen en accedir a una memòria.

### 5.1.5 Carrusel

El carrusel, després de diversos dissenys per saber on col·locar-lo, s'ha situat per sobre dels *frames Mirror*, sense sobreposar cap element. S'ha reduït la mida del carrusel i també la mida de les miniatures.

## 5.2 Altres modificacions

A més de totes les modificacions realitzades als diferents elements que formen l'aplicació també s'han produït altres modificacions generals, com les que s'expliquen a continuació.

S'ha implementat un disseny *responsive* per tal de ser visible i poder interactuar en diferents pantalles. La majoria d'elements tenen mides fixes per garantir una interacció correcta i, com a resultat, en pantalles amb resolucions molt antigues o no estàndards no ha estat possible adaptar l'aplicació completament.

S'ha afegit transparència als elements de control, és a dir, *frames Mirror, decks i sliders*. D'aquesta forma, el fons d'aquests elements no bloqueja part dels efectes produïts en els pòsters. Aquest problema no es produïa a l'aplicació original, ja que els pòsters i efectes es trobaven a la pantalla superior, aïllats de la resta d'elements.

S'han agregat una sèrie de vídeos de temàtica relacionada amb els pòsters i sons de l'aplicació per utilitzar-los com a salvapantalles després de trenta segons d'inactivitat. Aquests vídeos es reproduïen aleatòriament i en cicle fins que l'usuari no torna a mostrar activitat.

Finalment, s'han modificat les icones per poder eliminar un element audiovisual seleccionat prèviament.

### 5.3 Implementació d'una nova versió

En un punt del projecte, es va proposar la creació de versions diferents a *Memory Fields*, a més de l'adaptació que s'ha realitzat, utilitzant els elements que la formen. Es van generar dissenys com els mostrats a l'apartat 4.4 i es va implementar una d'aquestes versions, tot i que finalment es va decidir dedicar totalment el temps disponible a l'adaptació.

Es va implementar la versió del visor de col·leccions d'imatges. En aquest cas, es van utilitzar pòsters de pel·lícules com a exemple. Aquesta versió consta d'un carrusel amb pòsters sempre visible des d'on es pot seleccionar un d'ells. Aquest pòster es mostra en on àrea on pot realitzar zoom i moure la imatge mitjançant el *mouse*. Finalment es poden desar comentaris d'igual manera que a *Memory Fields*.

## 6 FASE DE PROVES

La fase de proves en l'àmbit de l'Enginyeria del Software es centra a comprovar que l'aplicació desenvolupada realitza correctament les tasques especificades a la fase de recollida de requisits.

En aquest projecte, s'han dut a terme proves unitàries per comprovar que el funcionament dels mètodes dels diferents elements de l'aplicació és l'adequat segons els requisits especificats. També s'han realitzat proves d'integració un cop s'han verificat les proves anteriors, per tal d'assegurar la compatibilitat entre tots els elements que formen l'aplicació. Finalment, s'han efectuat proves de sistema per verificar que l'aplicació realitza les funcions principals per les quals ha estat creada. El fet de provar les diferents parts del sistema ha permès garantir el comportament correcte de totes les funcionalitats.

Ja que s'ha seguit una metodologia àgil i iterativa, algunes de les proves esmentades s'han anat realitzant a mesura que s'avançava la implementació.

## 7 CONCLUSIONS

La realització de les diferents fases d'un projecte d'Enginyeria del Software i el fet de seguir una metodologia ha implicat el compliment de l'objectiu principal planificat

inicialment, és a dir, la creació d'una adaptació a partir de l'aplicació *Memory Fields*. El resultat de finalitzar aquest projecte no només ha estat l'adaptació creada, també s'han adquirit una sèrie de coneixements. S'ha après que el fet de realitzar un projecte basat en una aplicació existent implica una gran dedicació a l'estudi i comprensió d'aquesta. D'igual manera, s'han adquirit coneixements i experiència en les eines i llenguatges de desenvolupament que s'han utilitzat al llarg del projecte.

De cara al futur, ja que és una aplicació escalable, es podrien implementar funcionalitats completament noves o afegir nous elements per oferir noves i diferents experiències multimèdia als usuaris.

### AGRAÏMENTS

Personalment, vull agrair a Fernando Vilariño i a Dan Norton per donar-me la possibilitat de realitzar aquest projecte i ajudar-me amb reunions, opinions i consells a finalitzar-lo. També, dedicar un especial agraïment a Onur Ferhat per les explicacions i indicacions que m'ha proporcionat en tot moment que ho he necessitat.

### REFERÈNCIES

- [1] Internetlivestats.com. (2016). *Total number of Websites - Internet Live Stats*. [Online] Available at: <http://www.internetlivestats.com/total-number-of-websites/> [Accessed 25 Jun. 2016].
- [2] Norton, D., Vilariño, F. and Ferhat, O. (2015). *Memory Fields - a DJ in the Library*. [Online] [http://ablab.org/jump/image/isea/camp\\_de\\_mem.pdf](http://ablab.org/jump/image/isea/camp_de_mem.pdf). Available at: [http://ablab.org/jump/image/isea/camp\\_de\\_mem.pdf](http://ablab.org/jump/image/isea/camp_de_mem.pdf) [Accessed 25 Jun. 2016].
- [3] Boehm, B. (1988). *A Spiral Model of Software Development and Enhancement*. [Online] Available at: <https://www.dimap.ufrn.br/~jair/ES/artigos/SpiralModelBoehm.pdf> [Accessed 25 Jun. 2016].
- [4] Durfee, W. (2008). *Project Planning and Gantt Charts*. [online] Available at: [http://www.me.umn.edu/courses/me2011/handouts/proj\\_planning.pdf](http://www.me.umn.edu/courses/me2011/handouts/proj_planning.pdf) [Accessed 25 Jun. 2016].
- [5] Microsoftstore.com. (2016). *Microsoft Word - A word processor*. [Online] Available at: [https://www.microsoftstore.com/store/mseea/es\\_ES/pdp/Word-2016/productID.324453700](https://www.microsoftstore.com/store/mseea/es_ES/pdp/Word-2016/productID.324453700) [Accessed 25 Jun. 2016].
- [6] Cacao.com. (2016). *Cacao - A free online diagram tool*. [Online] Available at: <https://cacao.com/> [Accessed 25 Jun. 2016].
- [7] Adobe.com. (2016). *Adobe Photoshop CC - A raster graphics editor*. [Online] Available at: <http://www.adobe.com/es/products/photoshop.html> [Accessed 25 Jun. 2016].



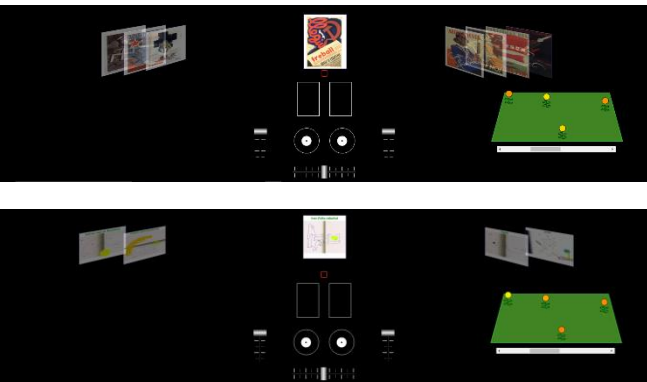
- [8] Djangoproject.com. (2016). *Django - The Web framework for perfectionists with deadlines*. [Online] Available at: <https://www.djangoproject.com/> [Accessed 25 Jun. 2016].
- [9] Sublimetext.com. (2016). *Sublime Text - A sophisticated text editor for code, markup and prose*. [Online] Available at: <https://www.sublimetext.com/> [Accessed 25 Jun. 2016].
- [10] GitHub. (2016). *GitHub - Build software better, together*. [Online] Available at: <https://github.com/> [Accessed 25 Jun. 2016].
- [11] Git-for-windows.github.io. (2016). *Git for Windows*. [Online] Available at: <https://git-for-windows.github.io/> [Accessed 25 Jun. 2016].
- [12] Openshift.com. (2016). *Red Hat's Platform-as-a-Service, to quickly develop, host, and scale applications in a cloud environment*. [Online] Available at: <https://www.openshift.com/> [Accessed 25 Jun. 2016].
- [13] Mysql.com. (2016). *MySQL - The world's most popular open source database*. [online] Available at: <https://www.mysql.com/> [Accessed 25 Jun. 2016].
- [14] Phpmyadmin.net. (2016). *PhpMyAdmin - A free software tool, intended to handle the administration of MySQL*. [Online] phpMyAdmin. Available at: <https://www.phpmyadmin.net/> [Accessed 25 Jun. 2016].
- [15] Ablab.org. (2016). *Installation of Memory Fields in Biblioteca de Sant Cugat, Volpelleres*. [Online] Available at: <http://ablab.org/jump/livlabin.html> [Accessed 25 Jun. 2016].
- [16] Ablab.org. (2016). *Installation of Memory Fields in Internet Librarian International 2015 Conference*. [Online] Available at: <http://ablab.org/jump/iliin.html> [Accessed 25 Jun. 2016].
- [17] Backbonejs.org. (2016). *Backbone.js - A structure to web applications*. [Online] Available at: <http://backbonejs.org/> [Accessed 25 Jun. 2016].
- [18] Grove, R. and Okzan, E. (2011). *The MVC-Web Design Pattern*. [Online] Available at: <http://grove.cs.jmu.edu/groverf/papers/WEBIST2011.pdf> [Accessed 25 Jun. 2016].
- [19] Lee, S. (2012). *Unified Modeling Language (UML) for Database Systems and Computer Applications*. [Online] Available at: [http://www.sersc.org/journals/IJDTA/vol5\\_no1/9.pdf](http://www.sersc.org/journals/IJDTA/vol5_no1/9.pdf) [Accessed 25 Jun. 2016].
- [20] Hooda, I. and Singh, R. (2015). *Software Test Process, Testing Types and Techniques*. [Online] Available at: <http://research.ijcaonline.org/volume111/number13/pxc3901433.pdf> [Accessed 25 Jun. 2016].
- [21] Socolofsky, S. (2004). *How to Write a Research Journal Article in Engineering and Science*. [Online] Available at: [https://ceprofs.civil.tamu.edu/ssocolofsky/downloads/paper\\_how-to.pdf](https://ceprofs.civil.tamu.edu/ssocolofsky/downloads/paper_how-to.pdf) [Accessed 25 Jun. 2016].

APÈNDIX

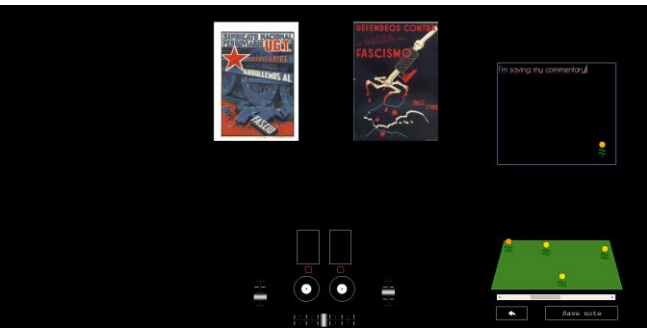
A1.1 Pantalla de l'adaptació en funcionament



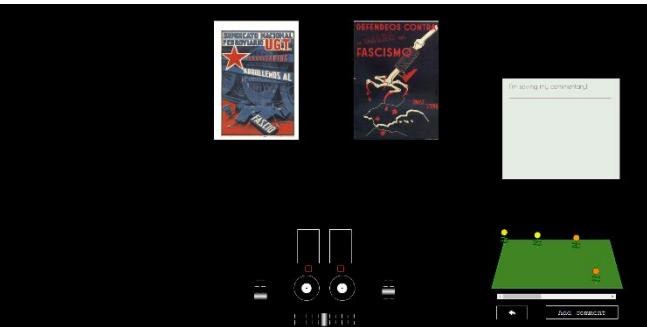
A1.2 Pantalla del carrusel de pòsters i sons



A1.2 Pantalla de la zona d'anotació



A1.3 Pantalla dels comentaris



A1.4 Exemple de salvapantalles



A1.5 Pantalla de la versió diferent implementada

